MISE: Providing Performance Predictability in Shared Main Memory Systems

Lavanya Subramanian, Vivek Seshadri, Yoongu Kim, Ben Jaiyen, Onur Mutlu (CMU)

PROBLEM

- Applications interfere at main memory
- Memory interference → different slowdowns for different applications

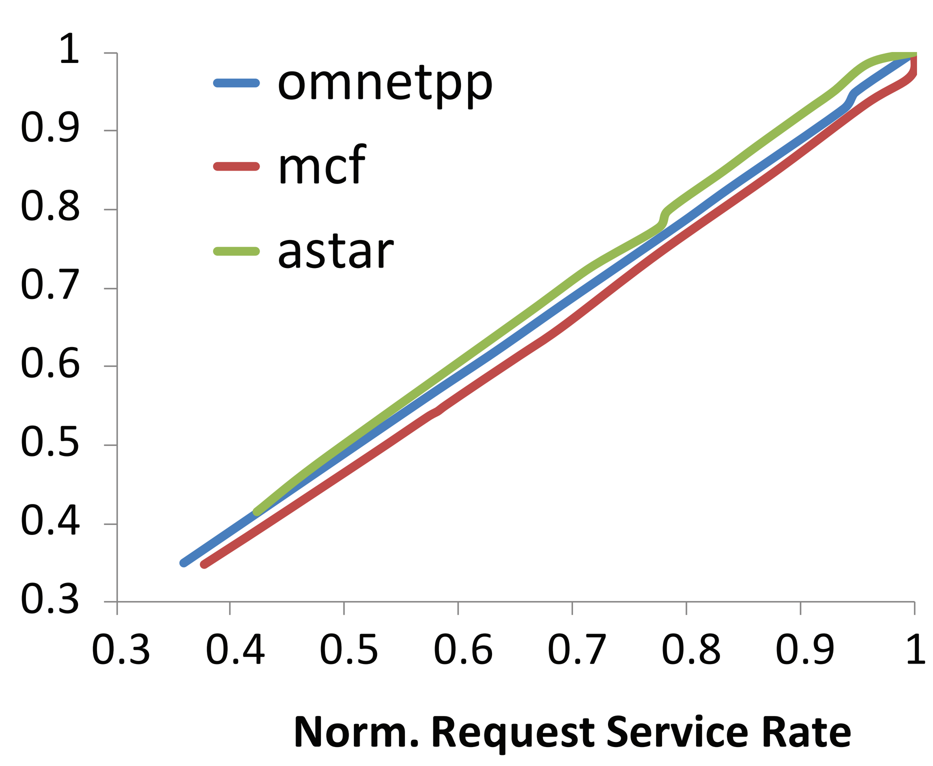
GOAL

Provide performance predictability using a simple and accurate slowdown estimation model

KEY OBSERVATIONS

OBSERVATION 1

- For a memory bound application, Performance α Request service rate



$$\text{Slowdown} = \frac{\text{Alone Request Service Rate (ARSR)}}{\text{Shared Request Service Rate (SRSR)}}$$

OBSERVATION 2

- ARSR measured by giving application highest priority
- Highest priority → Little interference

OBSERVATION 3

- Compute phase (1- α) does not slowdown due to memory interference

$$\text{Slowdown} = (1 - \alpha) + \alpha \frac{\text{ARSR}}{\text{SRSR}}$$

THE MISE MODEL

Measure SRSR

- Using performance counters

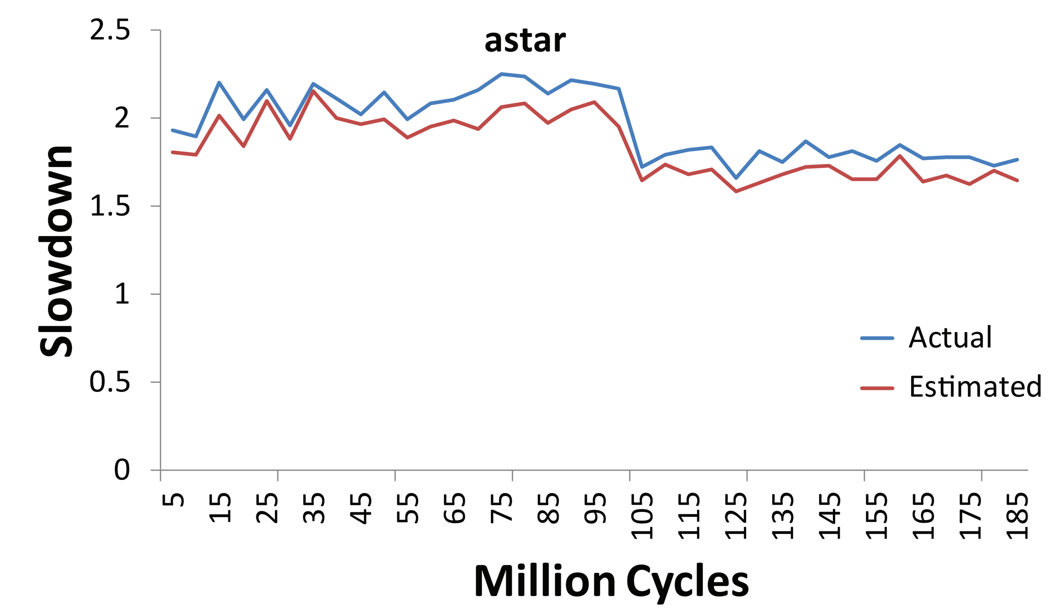
Estimate ARSR

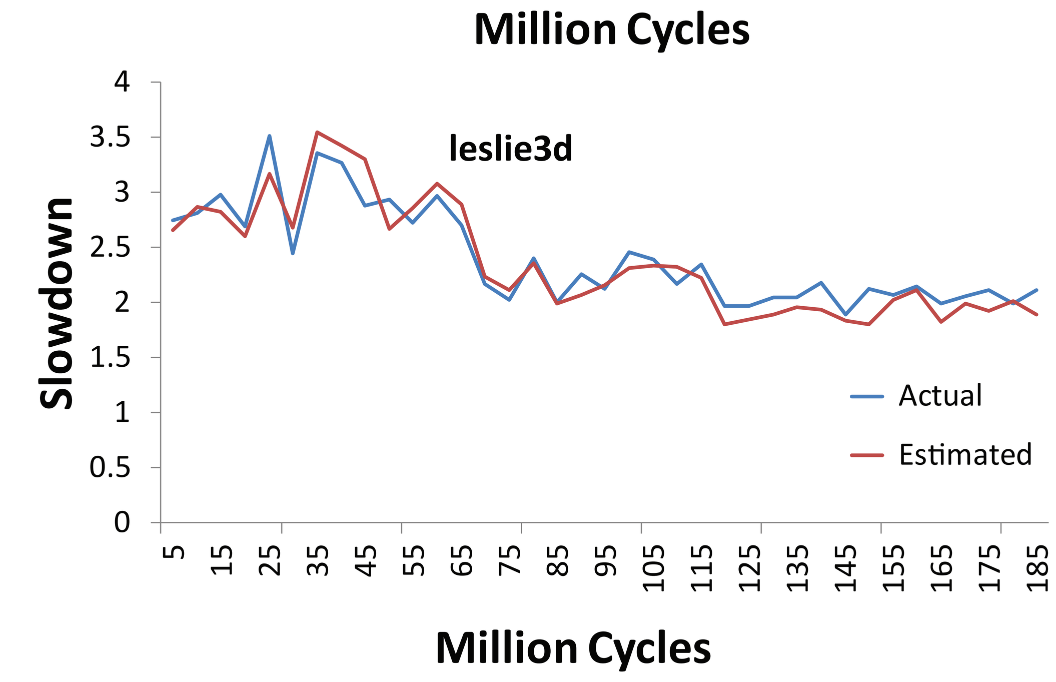
- Assign each application highest priority periodically
- Measure ARSR of an application when it has highest priority

Estimate Slowdown

- As function of ARSR, SRSR, α

MODEL: RESULTS





Average Error: 8.8 %
(across 300 workloads)

APPLICATIONS OF OUR MISE MODEL

PROVIDING SOFT QOS GUARANTEES

Goal

- Meet slowdown bound for QoS-critical applications
- Maximize system performance

Basic Idea

- Estimate slowdown using MISE model
- Just enough bandwidth to QoS-critical application to meet bound
- Spare bandwidth to other applications to improve performance

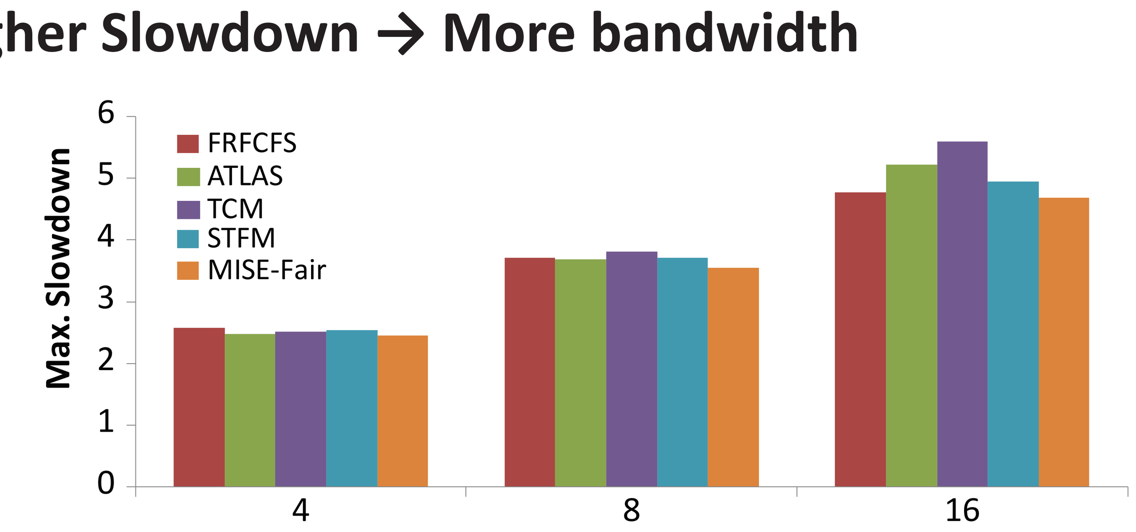
Results

- 4-core 1-channel system, 300 workloads
- Slowdown bound met for 90% workloads
- 10% better system perf. than always prioritizing QoS-critical application

IMPROVING SYSTEM FAIRNESS

Basic Idea

- Estimate slowdown using MISE model
- Higher Slowdown → More bandwidth



Intel Science & Technology Center for Cloud Computing

Carnegie Mellon University | Georgia Tech | intel | Princeton University | UC Berkeley | University of Washington